

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims**

1. (Previously Presented) A system for delivering an electronic document, comprising:

a transcoder proxy coupled to receive the electronic document in a first digital format, wherein the electronic document includes an element, and wherein the transcoder proxy is configured to:

assign a unique identifier to the element;

produce an original script that includes: (i) at least a portion of the electronic document expressed in a second digital format, and (ii) the element and the identifier assigned to the element; and

a client machine coupled to receive the original script and configured to:

use the original script to present the portion of the electronic document;

receive a modification script, which is produced by the transcoder proxy in response to an event generated at the client machine; and

use the modification script to modify the presented portion of the electronic document.

2. (Original) The system as recited in claim 1, wherein the transcoder proxy is further configured to store the element and the identifier assigned to the element for future use.

3. (Original) The system as recited in claim 1, wherein the first digital format is a text-based markup language.

4. (Original) The system as recited in claim 3, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).

5. (Canceled)

6. (Original) The system as recited in claim 1, wherein the second digital format is a scripting language.

7. (Original) The system as recited in claim 1, wherein the transcoder proxy is further configured to:

form a model of a logical structure of the electronic document;

use the model to produce the original script; and

provide the original script.

8. (Original) The system as recited in claim 7, wherein the model also defines methods for accessing and manipulating the document.

9. (Original) The system as recited in claim 8, wherein the model is a document object model (DOM).

10. (Canceled)

11. (Previously Amended) The system as recited in claim 7, wherein the client machine is further configured to:

generate the event in response to user input;

associate the event with the element within the original script; and

provide the event and the identifier assigned to the element associated with the event to the transcoder proxy.

12. (Previously Presented) The system as recited in claim 11, wherein elements of the electronic document are associated with corresponding identifiers within the model, and wherein in response to the event and the identifier provided by the client machine, the transcoder proxy is configured to:

access the model using the identifier;

use the model to produce the modification script, wherein the modification script differs from the original script; and

provide the modification script to the client machine.

13. (Canceled)

14. (Previously Presented) A system for delivering an electronic document, comprising:

a transcoder proxy coupled to receive the electronic document in a first digital format, wherein the electronic document includes at least one element, and wherein the transcoder proxy is configured to:

assign a unique identifier to each element of the electronic document;

form a model of a logical structure of the electronic document;

use the model to produce an original script, wherein the original script includes at least a portion of the electronic document expressed in a second digital format, and wherein the original script includes at least one element and the identifier assigned to the element; and

provide the original script; and

a client machine coupled to receive the original script, wherein the client machine is configured to:

use the original script to present the portion of the electronic document; and

receive a modification script, which is produced by the transcoder proxy in response to an event generated at the client machine, and use the modification script to modify the presented portion of the electronic document.

15. (Previously Presented) The system as recited in claim 14, wherein the client machine is further configured to:

generate the event in response to user input;

associate the event with the element within the original script; and

provide the event and the identifier assigned to the element associated with the event to the transcoder proxy.

16. (Original) The system as recited in claim 14, wherein the model also defines methods for accessing and manipulating the document.

17. (Original) The system as recited in claim 15, wherein the model is a document object model (DOM).

18. (Original) The system as recited in claim 14, wherein the first digital format is a text-based markup language.

19. (Original) The system as recited in claim 18, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).

20. (Previously Presented) The system as recited in claim 14, wherein elements of the electronic document are associated with corresponding identifiers within the model, and wherein in response to the event and the identifier provided by the client machine, the transcoder proxy is configured to:

access the model using the identifier;

use the model to produce the modification script, wherein the modification script differs from the original script; and

provide the modification script to the client machine.

21. (Canceled)

22. (Previously Presented) A transcoder proxy, comprising:

a synchronous document object model (DOM) generator adapted to receive an electronic document in a first digital format, wherein the electronic document includes at least one element, and wherein the synchronous DOM generator comprises an identifier (ID) generator configured to assign a unique identifier to each element, and wherein the synchronous DOM generator is configured to:

form a pre-transcoded DOM representing a logical structure of the electronic document, wherein elements are associated with corresponding identifiers within the pre-transcoded DOM; and

provide a first portion of the electronic document in the first digital format; and

a transcoder coupled to receive the portion of the electronic document in the first digital format and configured to:

translate the first portion of the electronic document from the first digital format to an original script in a second digital format, wherein the original script includes a element and the identifier assigned to the element;

provide the original script to a client machine coupled to the transcoder for presenting the first portion of the electronic document;

generate a modification script in response to user input at the client machine; and

provide the modification script to the client machine for modifying the presented portion of the electronic document.

23. (Original) The transcoder as recited in claim 22, wherein the synchronous DOM generator is further configured to:

receive DOM access commands;

access the pre-transcoded DOM using the DOM access commands; and

provide a second portion of the electronic document in the first digital format.

24. (Previously Presented) The transcoder as recited in claim 23, wherein the transcoder is further configured to:

receive the second portion of the electronic document in the first digital format; and

translate the second portion of the electronic document from the first digital format to the modification script in the second digital format.

25. (Original) The transcoder proxy as recited in claim 22, wherein the first digital format is a text-based markup language.

26. (Original) The transcoder proxy as recited in claim 25, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).

27. - 29. (Canceled)

30. (Previously Presented) A system for delivering an electronic document, comprising:

a transcoder proxy, including:

a synchronous document object model (DOM) generator coupled to receive the electronic document in a first digital format, wherein the electronic document includes at least one element, and wherein the synchronous DOM generator

comprises an identifier (ID) generator configured to assign a unique identifier to each element, and wherein the synchronous DOM generator is configured to:

form a pre-transcoded DOM representing a logical structure of the electronic document, wherein elements are associated with corresponding identifiers within the pre-transcoded DOM;

provide a portion of the electronic document in the first digital format;  
and

a transcoder coupled to receive the portion of the electronic document in the first digital format and configured to:

translate the portion of the electronic document from the first digital format to an original script in a second digital format, wherein the original script includes an element and the identifier assigned to the element; and

provide the original script; and

a client machine coupled to receive the original script, wherein the client machine comprises:

an output device;

a user agent coupled to the output device and the transcoder, wherein the user agent is configured to:

form a transcoded DOM in response to the original script from the transcoder, wherein the transcoded DOM is a representation of the portion of the electronic document;

use the transcoded DOM to produce output commands; and

provide the output commands to the output device.

31. (Previously Presented) The system as recited in claim 30, wherein the client machine is further configured to:

generate an event in response to user input;

associate the event with the element within the original script;

provide the event and the identifier assigned to the element associated with the event to a user interface generator coupled within the transcoder proxy; and

modify the transcoded DOM in response to a modification script received from the user interface generator.

32. (Canceled)

33. (Previously Presented) The method as recited in claim 37, wherein the model also defines methods for accessing and manipulating the document.

34. (Previously Presented) The method as recited in claim 37, wherein the model is a document object model (DOM).

35. (Previously Presented) The method as recited in claim 37, wherein the first digital format is a text-based markup language.

36. (Previously Presented) The method as recited in claim 37, wherein the second digital format is a scripting language.

37. (Previously Presented) A method for presenting an electronic document, comprising:

receiving the electronic document in a first digital format;

assigning a unique identifier to each element of the electronic document;



forming a model of a logical structure of the electronic document, wherein elements are associated with corresponding identifiers within the model;

using the model to produce an original script, wherein the original script includes at least a portion of the electronic document expressed in a second digital format, and wherein the original script includes at least one element and the identifier assigned to the at least one element;

providing the original script for use in presenting the portion of the electronic document;

receiving an event and an identifier associated with the event;

accessing the model using the identifier associated with the event;

using the model to produce a modification script, wherein the modification script differs from the original script; and

providing the modification script for use in modifying the presented portion of the electronic document.

38. (Canceled)

39. (Original) A method for presenting an electronic document, comprising:

receiving the electronic document in a first digital format;

assigning a unique identifier to each element of the electronic document;

forming a model of a logical structure of the electronic document, wherein elements are associated with corresponding identifiers within the model;

using the model to produce an original script, wherein the original script includes at least a portion of the electronic document expressed in a second digital format, and wherein the

original script includes an element of the document and an identifier assigned to the element;

using the original script to present the portion of the document;

generating an event in response to user input;

associating the event with the element within the original script;

accessing the model using the identifier;

using the model to produce modification script; and

using the modification script to modify the presented portion of the document.